

Федеральное государственное бюджетное образовательное учреждение высшего образования



«Московский государственный технический
университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ФУНДАМЕНТАЛЬНЫЕ НАУКИ

КАФЕДРА МАТЕМАТИКА И КОМПЬЮТЕРНЫЕ НАУКИ

Отчет

по лабораторному заданию № 11

Вариант 18

Дисциплина: Информатика

Название лабораторного задания: Использование указателей при работе с массивами в C++. Динамические одномерные массивы.

Студент гр. ФН11-22Б

Xp 02.03.21
(Подпись, дата)

М.Х. Хаписов
(И.О. Фамилия)

Преподаватель Доцент кафедры ИУ-6

T.H.
(Подпись, дата)

Т.Н. Ничушкина
(И.О. Фамилия)

02.03.2021

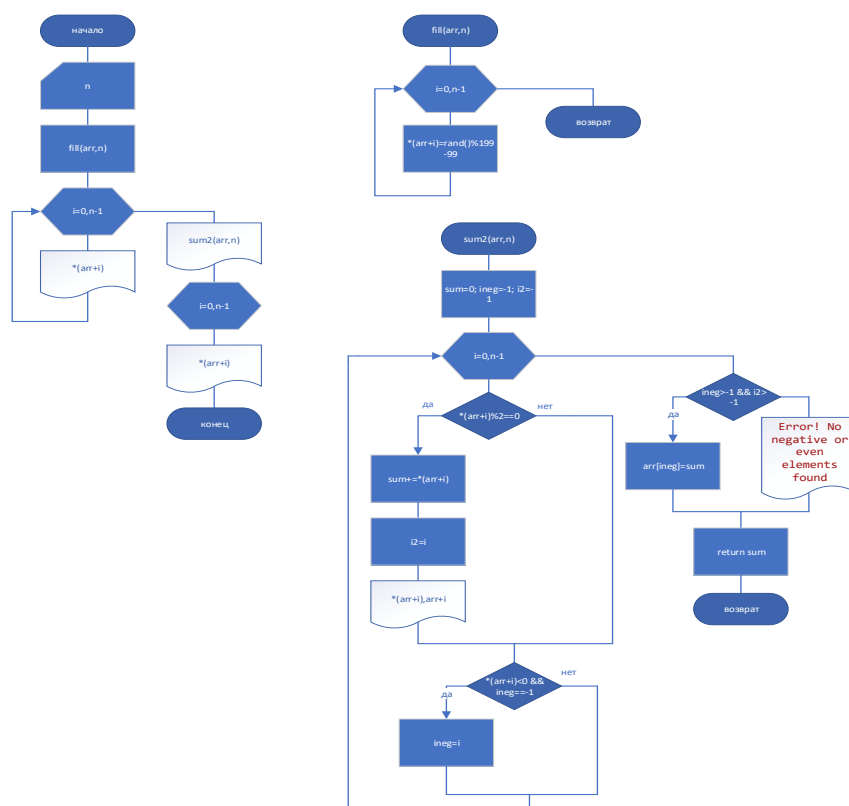
Москва, 2021

Цель:

- Приобретение навыков работы с динамической памятью и указателями на языке программирования C++.
- Изучение принципов и приемов работы с динамическими одномерными массивами.

Задание. Сформировать одномерный динамический целочисленный массив $A(n)$, (n вводится с клавиатуры). Определить сумму элементов массива, кратных двум, и поместить ее на место первого отрицательного элемента этого массива. Пользуясь **указателями**, вывести на экран адреса найденных элементов. Вывести на экран исходный и сформированный массивы, а также найденную сумму. Если отрицательных или кратных двум элементов в массиве нет – изменения не проводить и выдать соответствующее сообщение.

Схема алгоритма



Текст программы

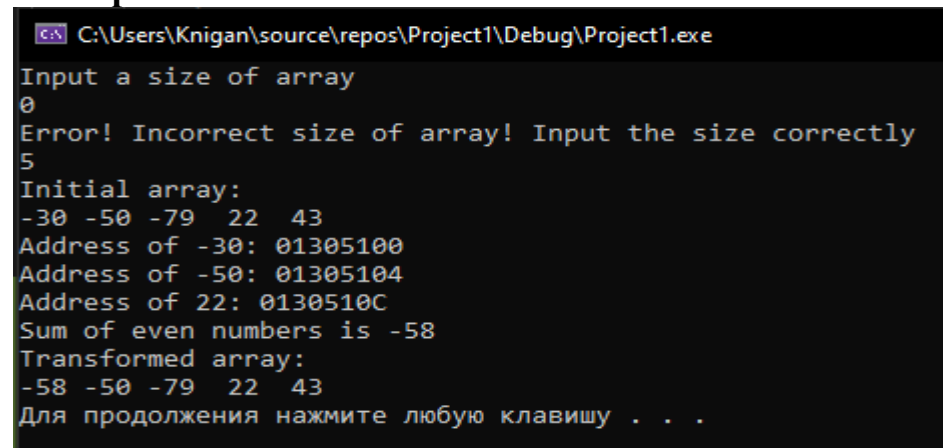
```
#include <iostream>
#include <ctime>
void fill(int *arr,int n) {
    srand(time(NULL));
```

```

        for (int i=0;i<n;++i) *(arr+i)=rand()%199-99;
    }
    int sum2(int *arr,int n) {
        int sum=0,ineg=-1,i2=-1;
        for (int i=0;i<n;++i) {
            if (*(arr+i)%2==0) {
                sum+=*(arr+i);
                i2=i;
                printf("Address of %d: %p\n",*(arr+i),arr+i);
            }
            if (*(arr+i)<0 && ineg==-1) ineg=i;
        }
        if (ineg>-1 && i2>-1) arr[ineg]=sum;
        else puts("Error! No negative or even elements found");
        return sum;
    }
    void main() {
        int n;
        puts("Input a size of array");
        std::cin>>n;
        while (n<1) {
            puts("Error! Incorrect size of array! Input the size correctly");
            std::cin>>n;
            int *arr=new int[n];
        }
        fill(arr,n);
        puts("Initial array:");
        for (int i=0;i<n;++i) printf("%3d ",*(arr+i));
        puts("");
        printf("Sum of even numbers is %d\n",sum2(arr,n));
        puts("Transformed array:");
        for(int i=0;i<n;++i) printf("%3d ",*(arr+i));
        puts("");
        system("pause");
    }
}

```

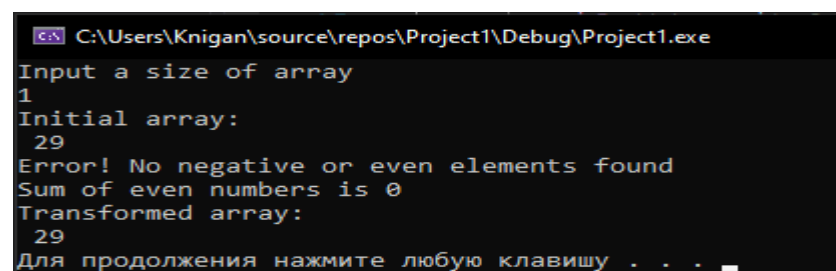
Тестирование



```

C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Input a size of array
0
Error! Incorrect size of array! Input the size correctly
5
Initial array:
-30 -50 -79 22 43
Address of -30: 01305100
Address of -50: 01305104
Address of 22: 0130510C
Sum of even numbers is -58
Transformed array:
-58 -50 -79 22 43
Для продолжения нажмите любую клавишу . . .

```



```

C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Input a size of array
1
Initial array:
29
Error! No negative or even elements found
Sum of even numbers is 0
Transformed array:
29
Для продолжения нажмите любую клавишу . . .

```

```

C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Input a size of array
2
Initial array:
20 16
Address of 20: 00A9E350
Address of 16: 00A9E354
Error! No negative or even elements found
Sum of even numbers is 36
Transformed array:
20 16
Для продолжения нажмите любую клавишу . . .

```

```

C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Input a size of array
2
Initial array:
-9 -19
Error! No negative or even elements found
Sum of even numbers is 0
Transformed array:
-9 -19
Для продолжения нажмите любую клавишу . . .

```

```

C:\Users\Knigan\source\repos\Project1\Debug\Project1.exe
Input a size of array
15
Initial array:
30 -15 57 -49 -97 53 -85 0 -3 -56 -11 -33 -81 -86 -5
Address of 30: 00A55450
Address of 0: 00A5546C
Address of -56: 00A55474
Address of -86: 00A55484
Sum of even numbers is -112
Transformed array:
30 -112 57 -49 -97 53 -85 0 -3 -56 -11 -33 -81 -86 -5
Для продолжения нажмите любую клавишу . . .

```

Заметим, что в последнем примере разность адресов последнего чётного элемента и первого чётного элемента составляет

$$00A55484 - 00A55450 = 34_{16} = 3 \cdot 16 + 4 = 52_{10}$$

Между элементами расположено 13 элементов типа `int` $13 \cdot \text{sizeof}(\text{int}) = 13 \cdot 4 = 52$
 Разница адресов совпадает с расстоянием между элементами, что и требовалось доказать.

Вывод: я приобрел некоторые навыки работы с динамической памятью и указателями на языке программирования С++ и изучил принципы и приёмы работы с динамическими одномерными массивами

Ответы на контрольные вопросы:

1. Что такое динамический одномерный массив? (Ответ: Динамический одномерный массив – это одномерный массив, размер которого может изменяться во время исполнения программы. Возможность изменения размера отличает динамический массив от статического, размер которого задаётся на момент компиляции программы)
2. Как динамически выделяется память под динамический одномерный массив?

```
int *DArray; // объявление указателя на целое число
```

```
DArray = new int [100]; // выделение памяти под массив на 100 элементов
```

3. Как выполнить доступ к элементам динамического массива?

```
(mass+i) ⇔ &(mass[i]) // адреса элементов
```

```
* (mass+i) ⇔ mass[i] // значения элементов
```

4. Как освобождается память, выделенная для динамического массива?

```
delete [] DArray; //освобождение памяти
```